

Integrantes del equipo: Emilio Cobos Benito, Javier Tomás Vicente, Juan Díez Martínez,

Reto Técnico Parte 2

1. Crear proyecto en Laravel 9.

Lo primero es crear un nuevo proyecto de Laravel con Composer.

```
emilio@emilio-virtualbox:/opt/lampp/rickymorty$ sudo composer create-project
laravel/laravel api
[sudo] contraseña para emilio:
Do not run Composer as root/super user! See https://getcomposer.org/root for
details
Continue as root/super user [yes]? yes
Creating a "laravel/laravel" project at "./api"
Installing laravel/laravel (v10.2.6)
- Installing laravel/laravel (v10.2.6): Extracting archive
```

2. Creación de la Base de Datos.


A continuación creamos la base de datos y editamos el .env agregando la base de datos que acabamos de crear.

```
emilio@emilio-virtualbox:/opt/lampp$ sudo /opt/lampp/bin/mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 38
Server version: 10.4.28-MariaDB Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement
.

MariaDB [(none)]> CREATE DATABASE rickymorty CHARACTER SET utf8 COLLATE utf8_
spanish_ci;
Query OK, 1 row affected (0,043 sec)
```



The screenshot shows a text editor window titled `/opt/lampp/rickymorty/api/.env`. The editor has a menu bar with `Archivo`, `Editar`, `Opciones`, `Buscar`, and `Ayuda`. Below the menu is a toolbar with icons for file operations and editing. The main text area contains the following configuration:

```
.env
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=rickymorty
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
```

3. Crear Modelo Personaje.

Creamos el modelo de nuestro crud que se llamara personaje:

```
emilio@emilio-virtualbox:/opt/lampp/rickymorty/api$ sudo php artisan make:model
el personaje -m

INFO Model [app/Models/personaje.php] created successfully.

INFO Migration [database/migrations/2023_09_15_095000_create_personajes_t
able.php] created successfully.
```

4. Crear controlador.

Creamos el controlador de nuestro proyecto:

```
emilio@emilio-virtualbox:/opt/lampp/rickymorty/api$ sudo php artisan make:con
troller Api/personajeController --api

INFO Controller [app/Http/Controllers/Api/personajeController.php] create
d successfully.
```

5. Configurar archivo de migración.

Configuramos nuestras migraciones:

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('personajes', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('status');
            $table->string('species');
            $table->string('gender');
            $table->string('url');

            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('personajes');
    }
};
```

6. Configurar Modelo Personajes.

```
Personaje.php ×
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Personaje extends Model
{
    use HasFactory;
    protected $fillable = ['name', 'status', 'species', 'gender', 'url'];
}
```

```
emilio@emilio-virtualbox:/opt/lampp/rickymorty/api$ sudo php artisan make:controller Api/personajeController --api
```

```
INFO Controller [app/Http/Controllers/Api/personajeController.php] created successfully.
```

```
emilio@emilio-virtualbox:/opt/lampp/rickymorty/api$ sudo php artisan migrate
[sudo] contraseña para emilio:
```

```
INFO Preparing database.
```

```
Creating migration table ..... 1,344ms DONE
```

```
INFO Running migrations.
```

```
2014_10_12_000000_create_users_table ..... 1,634ms DONE
```

```
2014_10_12_100000_create_password_reset_tokens_table ..... 2,555ms DONE
```

```
2019_08_19_000000_create_failed_jobs_table ..... 2,469ms DONE
```

```
2019_12_14_000001_create_personal_access_tokens_table ..... 2,568ms DONE
```

```
2023_09_15_095000_create_personajes_table ..... 2,527ms DONE
```

7. Configurar controlador.

```
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Personaje;

class PersonajeController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $personajes = Personaje::all();
        return $personajes;
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $personaje = new Personaje();

        $personaje->name = $request->name;
        $personaje->status = $request->status;
        $personaje->species = $request->species;
        $personaje->gender = $request->gender;
        $personaje->url = $request->url;
        $personaje->save();    }

    public function show(string $id)
    {
        $personaje = Personaje::find($id);
        return $personaje;    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, string $id)
    {
        $personaje = Personaje::findOrFail($request->id);
        $personaje->name = $request->name;
        $personaje->status = $request->status;
        $personaje->species = $request->species;
        $personaje->gender = $request->gender;
        $personaje->url = $request->url;
        $personaje->save();
        return $personaje;    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(string $id)
    {
        $personaje = Personaje::destroy($id);
        return $personaje;
    }
}
```

8. Crear las rutas en Laravel 9.

```
api.php x
<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Api\PersonajeController;

/*
|-----
| API Routes
|-----
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "api" middleware group. Make something great!
|
*/

Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});

Route::controller(PersonajeController::class)->group(function (){
    Route::get('/Personajes', 'index');
    Route::post('/Personaje', 'store');
    Route::get('/Personaje/{id}', 'show');
    Route::put('/Personaje/{id}', 'update');
    Route::delete('/Personaje/{id}', 'destroy');
});
```